An Adaptive Multigrid Barotropic Tropical Cyclone Track Model[*]

SCOTT R. FULTON[†]

Department of Mathematics and Computer Science
Clarkson University, Potsdam, NY 13699–5815

Submitted to *Monthly Weather Review*
December 1999

Abstract

This paper describes the application of adaptive multigrid techniques to the problem of tropical cyclone track prediction. Based on the nondivergent barotropic vorticity equation, the model uses an adaptive multigrid method to refine the mesh around the moving vortex. Like conventional nested-grid models, this model achieves nonuniform resolution by superimposing uniform grids of different mesh sizes. Unlike nested-grid models, multigrid processing uses the interplay between solutions on fine and coarse grids—in regions where they overlap—to: (1) solve the implicit problem for the streamfunction with optimum efficiency, (2) automatically achieve two-way interaction at the grid interfaces, and (3) provide accurate truncation error estimates for use in determining where to refine or coarsen the grids. An exchange rate algorithm accomplishes the latter task, approximately optimizing the grid selection based on a user-specified tradeoff between accuracy and computational work.

Numerical results demonstrate that the model chooses reasonable grids with minimal user intervention. Using adaptive mesh refinement is at least an order of magnitude more efficient than using a single uniform grid, and the overhead cost of adaptive regridding is less than two percent of the total execution time. The adaptive multigrid approach allows track prediction errors due to discretization to be essentially eliminated from the problem at a reasonable computational cost.

[*] Work supported by the Office of Naval Research under grants N00014-98-1-0103 and N00014-98-1-0368

[†] `http://www.clarkson.edu/~fulton`

1 Introduction

In many physical problems the spatial scale of the solution varies significantly over the domain. For example, a tropical cyclone is a relatively small-scale vortex embedded in a larger-scale flow, and accurate prediction of the vortex track may require resolving the flow both within and around the storm. Solving such problems numerically often requires variable resolution to accurately resolve the small-scale features without compromising the overall efficiency. Conventional *nested-grid techniques* (Kurihara et al., 1979; Berger and Oliger, 1984) achieve local mesh refinement by superimposing nested uniform grids of varying mesh sizes. Such methods have been widely used for many years (e.g., Kurihara and Bender, 1980; Juang and Hoke, 1992; Skamarock and Klemp, 1993). The success of conventional nested-grid tropical cyclone models such as the GFDL (Kurihara et al., 1998) and VICBAR (DeMaria et al., 1992) models demonstrates the value of this approach.

In contrast to nested-grid models, *multigrid methods* (Brandt, 1977) use multiple grids of different mesh size covering the *same* region. The goal here is to solve the discretized equations with optimum efficiency: using simple relaxation on each grid to smooth the error on the scale of that grid produces the solution (on the finest grid) to the level of truncation error in computational work proportional to the number of unknowns (usually in the work of just a few relaxation sweeps). In addition to speeding up the solution process, multigrid processing provides accurate truncation error estimates, which can be used to determine where to refine (or coarsen) the grids in a local refinement scheme or to achieve higher-order accuracy via extrapolation.

While the multigrid literature has many examples of combining multigrid processing with nested grids (e.g., Ciesielski et al., 1986; Bai and Brandt, 1987; Saleh, 1994; Ruge et al., 1995), in nested-grid meteorological models the interaction between the solution on different grid levels has in general not been exploited. Indeed, some of the complexity of the VICBAR model is due to the fact that the computational grids are specifically *not* allowed to overlap.

The purpose of this paper is to demonstrate the potential of adaptive multigrid methods for time-dependent problems requiring local mesh refinement by describing an adaptive multigrid tropical cyclone track model. In the tradition of the barotropic models SANBAR (Sanders et al., 1975) and VICBAR, we refer to this multigrid barotropic model as MUDBAR. Our primary focus here is on the numerical method: describing the adaptive multigrid approach in detail and documenting its performance. Indeed, the MUDBAR model currently has the simplest possible dynamics (nondivergent barotropic equation) and no physics. Nevertheless, the model is fast and accurate, and could be useful for operational forecasting (in particular, ensemble forecasts) if augmented with boundary data from a global model and an appropriate initialization scheme.

The remainder of the paper is organized as follows. Section 2 describes the model equations and solution method. The grid adaptation algorithm is presented in section 3. Numerical results appear in section 4, and our conclusions are summarized in section 5.

2 Model Description

In this section we briefly describe the governing equations, discretization, grid structure, and solution method of the MUDBAR model. A preliminary version of the model was described in Fulton (1997), to which the reader is referred for some of the more basic details.

2.1 Governing Equations

We formulate the model on a section of the sphere, transforming longitude λ and latitude ϕ to Cartesian coordinates x and y via the Mercator projection

$$x = (\lambda - \lambda_c) a \cos\phi_c, \qquad y = \left[\tanh^{-1}(\sin\phi) - \tanh^{-1}(\sin\phi_c)\right] a \cos\phi_c, \tag{1}$$

where a is the radius of the earth, so the projection is true at (λ_c, ϕ_c) where $(x, y) = (0, 0)$. The model consists of the modified barotropic vorticity equation

$$\frac{\partial\zeta}{\partial t} + m^2 \frac{\partial(\psi, \zeta)}{\partial(x, y)} + \beta m \frac{\partial\psi}{\partial x} = \nu m^2 \nabla^2 \zeta, \tag{2}$$

where the relative vorticity ζ and streamfunction ψ are related by

$$\left(m^2 \nabla^2 - \gamma^2\right) \psi = \zeta. \tag{3}$$

Here $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$, $\beta = 2\Omega a^{-1} \cos\phi$ (with Ω the rotation rate of the earth), and $m = \cos\phi_c / \cos\phi$ is the map factor. The β-plane approximation is recovered by setting $m = 1$ and $\beta = 2\Omega a^{-1} \cos\phi_c$. There are two quasi-physical parameters: the diffusion coefficient ν, and the parameter γ (inverse of the effective Rossby radius) which helps prevent retrogression of ultralong Rossby waves. The model domain is a rectangle in x and y centered at $(x, y) = (0, 0)$. At the boundaries we specify the streamfunction ψ (and thus the normal component of the velocity); where there is inflow, we also specify the vorticity ζ. With these conditions the problem is well-posed (Oliger and Sundström, 1978).

2.2 Discretization

The space discretization uses second-order finite differences on uniform rectangular grids (as detailed below), approximating the advection terms by the Arakawa Jacobian (Arakawa, 1966). Details of the boundary discretization are given in Fulton (1997). On a uniform grid with mesh size h in x and y, the space-discretized approximations to (2) and (3) can be represented in the form

$$\frac{d\zeta^h}{dt} = F^h(\psi^h, \zeta^h) \tag{4}$$

and

$$L^h \psi^h = \zeta^h, \tag{5}$$

where the grid functions ψ^h and ζ^h consist of the values of the approximate solution on the grid h.

The time discretization uses the classical fourth-order Runge-Kutta (RK4) scheme; this is highly accurate, allows relatively large time steps for stability, and—since it is a one-step scheme—is easy to implement and has no spurious computational modes. This scheme involves four stages per time step, each of which predicts a new value for ζ^h via (4) and solves (5) for the corresponding ψ^h.

2.3 Grid Structure and Local Time Stepping

The above discretizations are embedded in an adaptive method which includes local refinements in both space and time, superimposing nested uniform grids with different mesh sizes to adapt the resolution near the storm. The base grid G_1 with mesh size h_1 covers the entire computational domain, while successively finer patches G_l with mesh sizes h_l ($l = 2, \ldots, n$) cover smaller nested areas as shown in the example in Fig. 1. We use the mesh ratio $h_{l-1}/h_l = 2$ (to facilitate multigrid processing), and require the patches to be *strictly nested* (fine grid contained in the interior of the next coarser grid) and *aligned* (patch boundaries coincide with coarse-grid lines). For simplicity, we allow only one region of refinement (i.e., one grid patch per mesh size), but this is not a fundamental limitation of the method.

The time stepping algorithm uses *local time stepping*, i.e., smaller time steps on finer grids; this algorithm is similar to that used in most nested-grid models (e.g., Berger and Oliger, 1984; DeMaria et al., 1992). Specifically, with two grids (coarse and fine) one full time step is executed as follows:

1. One step (length Δt) on the coarse grid,
2. Two steps (length $\Delta t/2$) on the fine grid, using boundary values interpolated from the coarse grid in space (ζ^h linearly and ψ^h by cubic interpolation) and time (linear interpolation),
3. Transfer the fine-grid solution to the coarse grid where they overlap (ψ^h by injection and ζ^h by full weighting).

This algorithm generalizes recursively to more than two grids. Its application to three computational grids is illustrated in Fig. 2.

2.4 Multigrid solution for streamfunction

To solve (5) efficiently for the streamfunction ψ^h on any computational grid G_l we use a multigrid method. This uses point Gauss-Seidel relaxation with red-black ordering, full

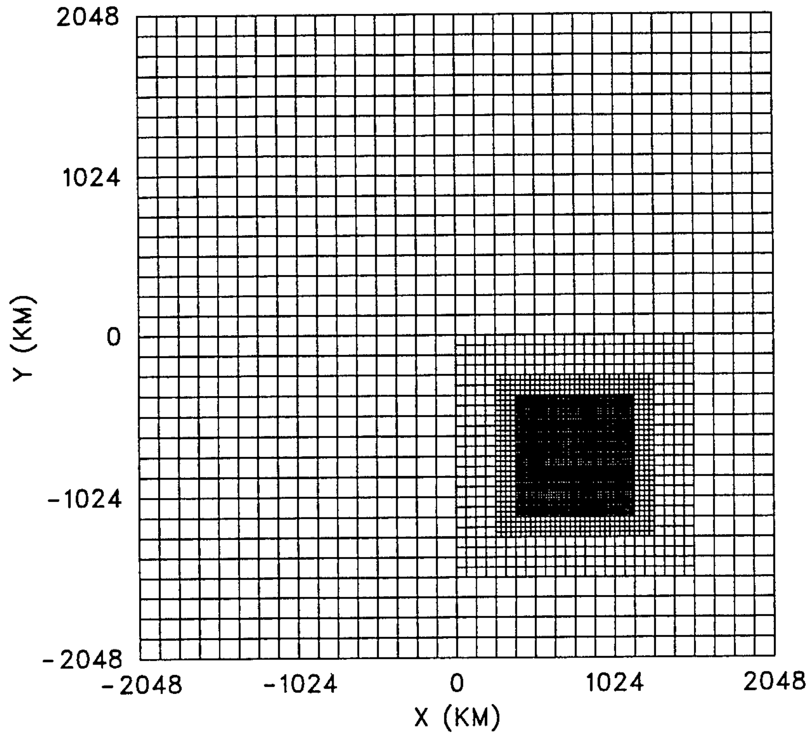

Figure 1: Sample computational grids ($h = 128$, 64, 32, 16 km).

weighting of residuals, bilinear interpolation of corrections, and the full multigrid (FMG) algorithm with one V(1,1)-cycle per level and bicubic initial interpolation. As these elements are all standard, the reader is referred to Fulton et al. (1986) and the references therein for details. The resulting algorithm solves accurately for the streamfunction, producing residuals smaller than the truncation error; the computational work is comparable to about eight relaxation sweeps on the finest grid, and overall accounts for about two-thirds of the total execution time of the model[1]. Numerical results show that using more sweeps per cycle (or more cycles) produces smaller residuals but no significant improvement in the track error, and thus is not worth the extra computational expense.

When used in the context of the adaptive grid structure described above, the standard multigrid approach is modified in several ways which deserve mention. First, with local time stepping, the next coarser grid G_{l-1} may be at a different time t (e.g., at the end of time steps 2, 3, and 6 in Fig. 2). In this situation the main computational grids cannot be used for multigrid processing. Instead, for each computational grid G_l we use an additional set

[1] Achieving the same accuracy with single-grid relaxation (SOR with optimal relaxation parameter) requires about 50–100 sweeps and roughly ten times as much total computer time.

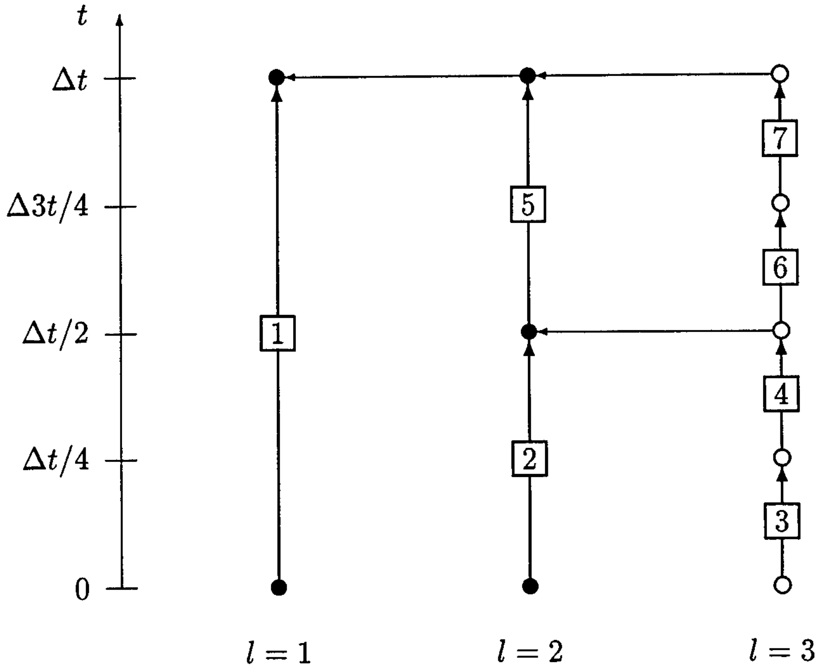

Figure 2: Example of time stepping algorithm for one full time step Δt on three computational grids. The steps are executed in the numbered order, with left arrows denoting the transfer of the solution from the fine grid to the coarse grid where they overlap. Solid circles mark times and grids where the truncation error is estimated for regridding.

of coarse grids with mesh sizes $2h_l$, $4h_l$, $8h_l$, etc., covering the same domain as G_l (Fulton, 1997). These "local coarse grids" may be used in two ways. The Berger-Oliger (BO) method uses them at all times, resulting in only one-way interaction between the grids: the fine grid uses boundary values from the coarse grid (step 2 above), but its effect is felt on the coarse grid only through the solution transfer in the overlap region at the end of the time step (step 3 above). In contrast, the multigrid (MG) method uses the full computational grid G_{l-1} instead of the first local coarse grid during multigrid cycling in the final RK4 stage of the second fine-grid time step (e.g., at the end of time steps 4, 5, and 7 in Fig. 2). This results in automatic two-way interaction between the computational grids, since fine-grid information can affect the entire coarse grid immediately through the relaxation process. Preliminary results (Fulton, 1997) showed that the differences between these two methods are generally small; further experimentation (see section 4.4) shows some improvement with the MG approach, so that is used in all other results presented here.

Second, using the full computational coarse grid in the MG method requires Full Approximation Scheme (FAS) processing (Brandt, 1977), since each grid patch covers a smaller domain than the previous one. Specifically, if on a grid patch G_l ($l > 1$) with mesh size $h = h_l$ we denote by $\tilde{\psi}^h$ the current approximation to the true (discrete) solution ψ^h, then the diagnostic equation to be solved in the overlap region on the next coarser grid G_{l-1} with mesh size $2h = h_{l-1}$ is

$$L^{2h}\hat{\psi}^{2h} = \hat{\zeta}^{2h} := L^{2h}(\hat{I}_h^{2h}\tilde{\psi}^h) + I_h^{2h}(\zeta^h - L^h\tilde{\psi}^h), \tag{6}$$

and the corresponding correction to the fine-grid solution is given by

$$\tilde{\psi}^h \leftarrow \tilde{\psi}^h + I_{2h}^h(\hat{\psi}^{2h} - \hat{I}_h^{2h}\tilde{\psi}^h). \tag{7}$$

Here, the fine-to-coarse transfer operators $\hat{I}_h^{2h}$ and I_h^{2h} represent injection and full weighting, respectively, and the coarse-to-fine transfer operator I_{2h}^h represents bilinear interpolation. In the region of the coarse grid not covered by the fine grid, the FAS equation (6) is replaced by $L^{2h}\psi^{2h} = \zeta^{2h}$ and no correction (7) is needed. When the local coarse grids are used (e.g., for the first time step on the fine grid) FAS processing is still used for convenience, even though the simpler correction scheme could be used.

Third, to properly represent the net fine-grid vorticity on the coarse grid when the grids are nested, a correction must be applied at the grid interfaces. As shown by Bai and Brandt (1987), this correction is accomplished by setting the coarse-grid vorticity at the grid interface to be

$$\zeta^{2h} := B_h^{2h}\left(\zeta^h + \frac{D_n^{2h}\tilde{\psi}^h}{2h}\right) - \frac{D_n^{2h}\hat{I}_h^{2h}\tilde{\psi}^h}{2h}. \tag{8}$$

Here B_h^{2h} is the one-dimensional full weighting operator along the boundary and D_n^{2h} is the one-sided difference approximation (over grid length $2h$) to the outward normal derivative[2]. For example, at a point along the eastern boundary of a fine-grid patch indexed in x and y as (b, j) on the fine grid and (B, J) on the coarse grid, we have

$$(D_n^{2h}\tilde{\psi}^h)_{b,j} = \frac{\tilde{\psi}_{b,j}^h - \tilde{\psi}_{b-2,j}^h}{2h}, \qquad (D_n^{2h}\tilde{\psi}^{2h})_{B,J} = \frac{\tilde{\psi}_{B,J}^{2h} - \tilde{\psi}_{B-1,J}^{2h}}{2h}, \tag{9}$$

and

$$(B_h^{2h}\zeta^h)_{B,J} = \tfrac{1}{4}\zeta_{b,j-1}^h + \tfrac{1}{2}\zeta_{b,j}^h + \tfrac{1}{4}\zeta_{b,j+1}^h, \tag{10}$$

where $\tilde{\psi}^{2h} = \hat{I}_h^{2h}\tilde{\psi}^h$ is the initial FAS approximation on the coarse grid. Other than the correction (8), no special treatment is required at the grid interfaces; in particular, no extended overlap or transition region is used or needed.

[2] In Bai and Brandt (1987) the sign of D_n^{2h} is reversed, which appears to be an error.

3 The Grid Adaptation Algorithm

The preliminary version of the model described in Fulton (1997) used *movable grids*, i.e., patches with sizes fixed in advance were simply moved when necessary to keep them approximately centered on the vortex. The results indicated that some choices of patch sizes substantially increased the accuracy or efficiency (relative to using a single uniform grid), while others led to significantly less improvement. Since the optimal combination of patch sizes will depend on both the vortex and the surrounding flow and may vary as the solution evolves, choosing fixed patch sizes "manually" is less than ideal: a fully automatic algorithm would be advantageous. This section details such an algorithm; we refer to the grids it chooses as *adaptive grids.*

3.1 Basic regrid strategy

The MUDBAR model *regrids*, i.e., chooses patch sizes and locations adaptively, by estimating the truncation error and placing the grid patches where the truncation error is large. The regridding algorithm is embedded in the local time stepping algorithm of section 2.3 as follows (see Fig. 3). Before starting a time step on a given grid with mesh size h—here considered to be the "coarse grid"—the truncation error is estimated (see section 3.2). Points where the truncation error is large (see section 3.3) are flagged for refinement, as indicated by "$\times$" in Fig. 3(a). Second, after the the time step is complete the truncation error is again estimated and large values are flagged for refinement, as shown in Fig. 3(b). Third, the smallest acceptable patches at the beginning and end of the step are determined (dotted and dashed outlines, respectively, in Fig. 3), and a suitable fine-grid patch with mesh size $h/2$ is chosen to cover both of these (solid outlines in Fig. 3(b)). Finally, that patch is "filled" with values of the vorticity $\zeta^{h/2}$ and streamfunction $\psi^{h/2}$. The patch so constructed is then used for the two fine-grid time steps which coincide with the single time step originally taken on the coarse grid.

This process is repeated recursively, i.e., during each of the two fine-grid steps a size and location are chosen for the next finer patch (if needed), proceeding to increasingly finer grids until either further refinement is not needed or a specified maximum number of grids is reached. This recursion is illustrated in Fig. 2, where the solid circles denote times at which the truncation error is estimated. For example, the truncation error on level $l = 2$ is computed before and after time step 5 (at $t = \Delta t/2$ and $t = \Delta t$, respectively) and used to determine the location and size of the next finer patch $l = 3$ for time steps 6 and 7 (from $t = \Delta t/2$ to $t = \Delta t$).

To fill a given patch, the vorticity $\zeta^{h/2}$ is set by copying values from the previous version of that grid (if any) where they coincide and using bilinear interpolation from the coarse grid where the fine grid is new; at $t = 0$ the specified initial vorticity is used instead. The streamfunction $\psi^{h/2}$ is set likewise except that bicubic interpolation is used; if the patch is new (e.g., at $t = 0$) then $\psi^{h/2}$ is obtained instead by solving (5) with boundary values

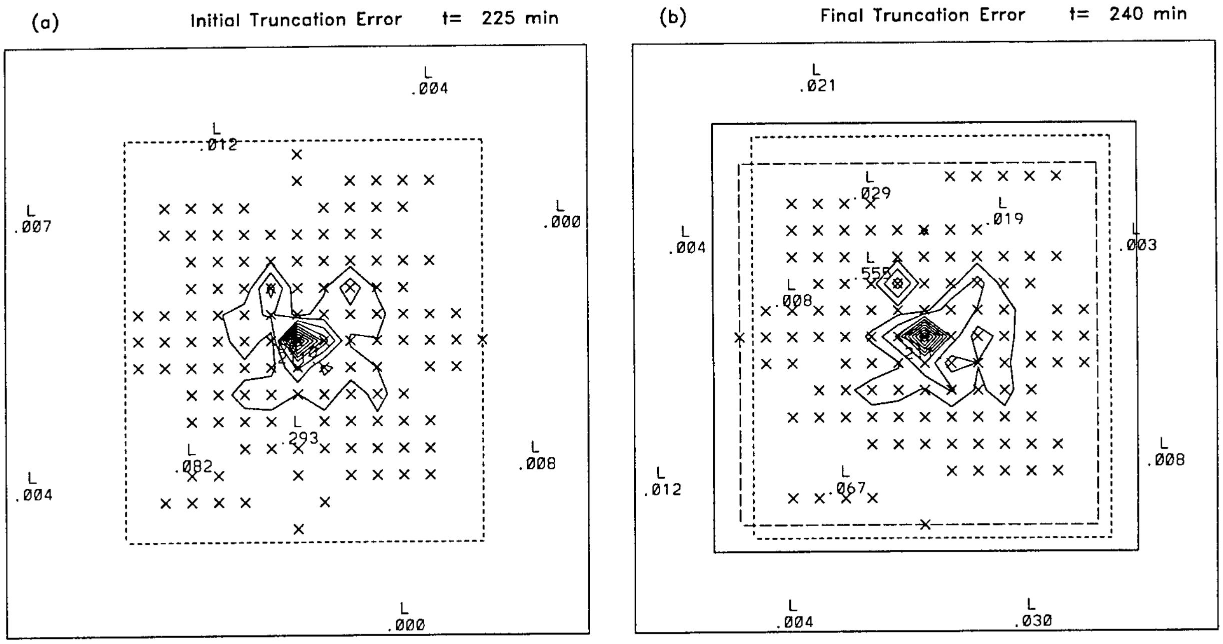

Figure 3: Choice of a fine-grid patch. The contours show truncation errors (large values flagged by ×) at: (a) the start of the time step, and (b) the end of the time step. The boundaries of the minimum acceptable patch at the start (dotted) and end (dashed) are shown, along with the patch chosen for the step (solid).

interpolated (cubically) from the coarse grid. During model initialization at $t = 0$, adaptive grids are constructed by starting with a specified base grid and constructing patches as dictated by the truncation error as outlined above. If the first patch covers most of the base grid (i.e., the base grid resolution is inadequate), this patch is extended to the full domain to replace the base grid and the process is repeated. Once the initial grids are set—and thus the fine-grid information has influenced the solution on all coarser levels–a second pass through the grid adaptation algorithm is made to take advantage of that fine-grid information.

Even though new grids are chosen for each time step on each level, the regridding process adds surprisingly little overhead to the execution time. All calculations required for regridding (estimating the truncation error, choosing the grid, and filling it by copying and/or interpolation) typically amount to less than two percent of the total execution time of the model. This slight overhead is more than made up for by the fact that the grid chosen at each step on each level is precisely the size needed for the entire step, and no larger. Indeed, a simpler scheme based on regridding on all levels only at the end of each base-grid time step regrids much less often but must choose larger patches (since they must be adequate for several fine-grid time steps) and thus takes more time (typically 20–30% more).

The model also includes an option for using movable grids, i.e., fixed patch sizes specified in advance. This differs from the preliminary version described in Fulton (1997) in that the patch locations are now chosen for an interval in time, rather than an instant. This is done by finding the vortex center at the beginning and end of the coarse-grid time step and centering the fine-grid patch for the corresponding two fine-grid time steps halfway between these two points. Except for this choice of patch location (and the fixed size), the regrid strategy with movable grids is identical to that described above for adaptive grids.

3.2 Truncation error estimates

The truncation error estimate needed for grid adaptation may be computed essentially for free using FAS processing; this idea was introduced by Brandt (1977) and is similar to that described by Skamarock (1989). Specifically, we want to base our grid refinement criterion on the truncation error

$$\tau^h := L^h \hat{I}^h \psi - \hat{I}^h \zeta, \tag{11}$$

where ψ and ζ represent the true solution of the continuous problem (3) and $\hat{I}^h$ represents their pointwise restriction to grid h. It can be shown that the *relative truncation error*

$$\tau_h^{2h} := \zeta^{2h} - \hat{I}_h^{2h} \zeta^h \tag{12}$$

provides an accurate approximation to the truncation error difference $\tau^{2h} - I_h^{2h} \tau^h$. In contrast to the analysis of Bernert (1997), we find that with injection as the fine-to-coarse transfer $\hat{I}_h^{2h}$ in (12), this estimate is fourth-order accurate (e.g., Fulton, 1989). Since $\tau^h = O(h^2)$, we obtain the estimate

$$\tau^h \approx \frac{1}{3} \tau_h^{2h} \tag{13}$$

which has accuracy $O(h^4)$. Since ζ^{2h} is known only at the coarse-grid points, it is possible to compute τ_h^{2h} only at the coarse-grid points; however, the estimate (13) is scaled so it is indeed an estimate of the truncation error on the fine grid. Note that while τ_h^{2h} can be computed from (12) with essentially no work (only one subtraction per coarse-grid point) during the last V-cycle of the FMG method, computing it separately after completing that cycle adds very little work and produces a better estimate of τ^h, since it is based on the final solution for ψ^h (rather than the initial approximation).

3.3 Patch location and size

The determination of what constitutes a "large" value of truncation error follows the method outlined by Brandt (1977), which uses a parameter λ, called the *exchange rate*, to control the trade-off between increased accuracy and increased work[3]. A point on the grid h is flagged for refinement if

$$h^2 \tau^h \geq \lambda \tag{14}$$

[3] The exchange rate λ originates as a Lagrange multiplier in the problem of minimizing the error subject to the constraint of constant work (or minimizing the work subject to constant error).

there. In this criterion, h^2 accounts for the work needed on the new grid and τ^h measures the accuracy. Thus, using a large value of λ says to refine only when the error is large, and thus should give a fast solution—with a relatively large error. Conversely, using a small value of λ indicates a willingness to exchange more work to get more accuracy (smaller error).

Once the points where the truncation error on the coarse grid (mesh spacing h) is large—in the sense of (14)—are located, the boundaries of the fine-grid patch (mesh spacing $h/2$) must be determined. For simplicity, the model currently uses only rectangular grids (and only one grid per value of h, so only one region is refined), so we can simply determine the smallest rectangles enclosing the large truncation errors before and after the step, as shown in Fig. 3. The tentative boundary of the patch for the full time step is simply the smallest rectangle enclosing both of these (i.e., their convex hull). This boundary may then be adjusted by adding a "buffer" of several grid intervals (usually two) to give some flexibility when the truncation error is highly localized. Finally, the patch size may be increased (or possibly slightly decreased) if needed to achieve good *coarsenability*, i.e., so that the number of points on the coarsest local coarse grid is small, thus increasing the efficiency of solving for the streamfunction on the patch. The algorithm employed here uses simple work estimates as detailed in the Appendix. In any case, strict nesting of the patch is enforced.

4 Results

This section summarizes the results of many model runs designed to explore the performance of the model for various situations.

4.1 Initial conditions

Except as otherwise noted, we use the initial conditions of DeMaria (1985) and associated parameter values as follows. The initial vortex has tangential wind given by

$$V(r) = 2V_m \left(\frac{r}{r_m}\right) \frac{\exp[-a(r/r_m)^b]}{1+(r/r_m)^2}, \tag{15}$$

where $r = [(x-x_0)^2 + (y-y_0)^2]^{1/2}$ is the radial distance from the vortex center (x_0, y_0). Note that V has the approximate maximum value V_m near $r = r_m$ (exact when $a = 0$); the exponential factor is included to make V vanish quickly for large r. The vortex can be made slightly elliptical (to introduce some small-scale structure, as in the spiral bands studied by Guinn and Schubert (1993)) by dividing the x or y factor in the definition of r by $(1-e^2)$, where e is the desired ellipticity. We consider two cases: a "weak hurricane" with $V_m = 30\,\mathrm{m\,s^{-1}}$ and $r_m = 80\,\mathrm{km}$, and a "strong hurricane" with $V_m = 60\,\mathrm{m\,s^{-1}}$ and $r_m = 20\,\mathrm{km}$. For both we use the values $a = 10^{-6}$, $b = 6$, and $e = 0.8$. The environmental flow is the zonal current with streamfunction given by

$$\bar{\psi}(y) = \left(\frac{\bar{u}_0 L}{2\pi}\right) \cos\left(\frac{2\pi y}{L}\right). \tag{16}$$

The computational domain is a square of side length 4096 km on a β-plane centered at latitude 20° N. The vortex is centered initially at $x_0 = 768\,\mathrm{km}$ and $y_0 = -768\,\mathrm{km}$, and we use the values $\bar{u}_0 = 10\,\mathrm{m\,s^{-1}}$, $L = 4096\,\mathrm{km}$, and $\gamma = 0$ and $\nu = 0$.

4.2 Sample solution

Figure 4 illustrates the solution of the model for the weak hurricane case. The left-hand panels show the streamfunction ψ and the right-hand panels show the corresponding vorticity ζ. For this run, the base grid has size 64×64 ($h_1 = 64\,\mathrm{km}$), the exchange rate is $\lambda = 1000$, and the grid refinement is limited to three patches, so the finest mesh used is $h_4 = 8\,\mathrm{km}$. The boundaries of the grid patches chosen by the adaptive algorithm are shown in the figures. It can be seen that the patch sizes tend to grow slightly during the model run; the apparent slight bias toward refining the grids behind the vortex is likely due to the larger truncation errors in the wake which are characteristic of centered finite differences. This effect could probably be lessened by including a small amount of dissipation; however, the model run shows that no dissipation is needed to achieve reasonable performance. In particular, no problems are observed at the grid interfaces.

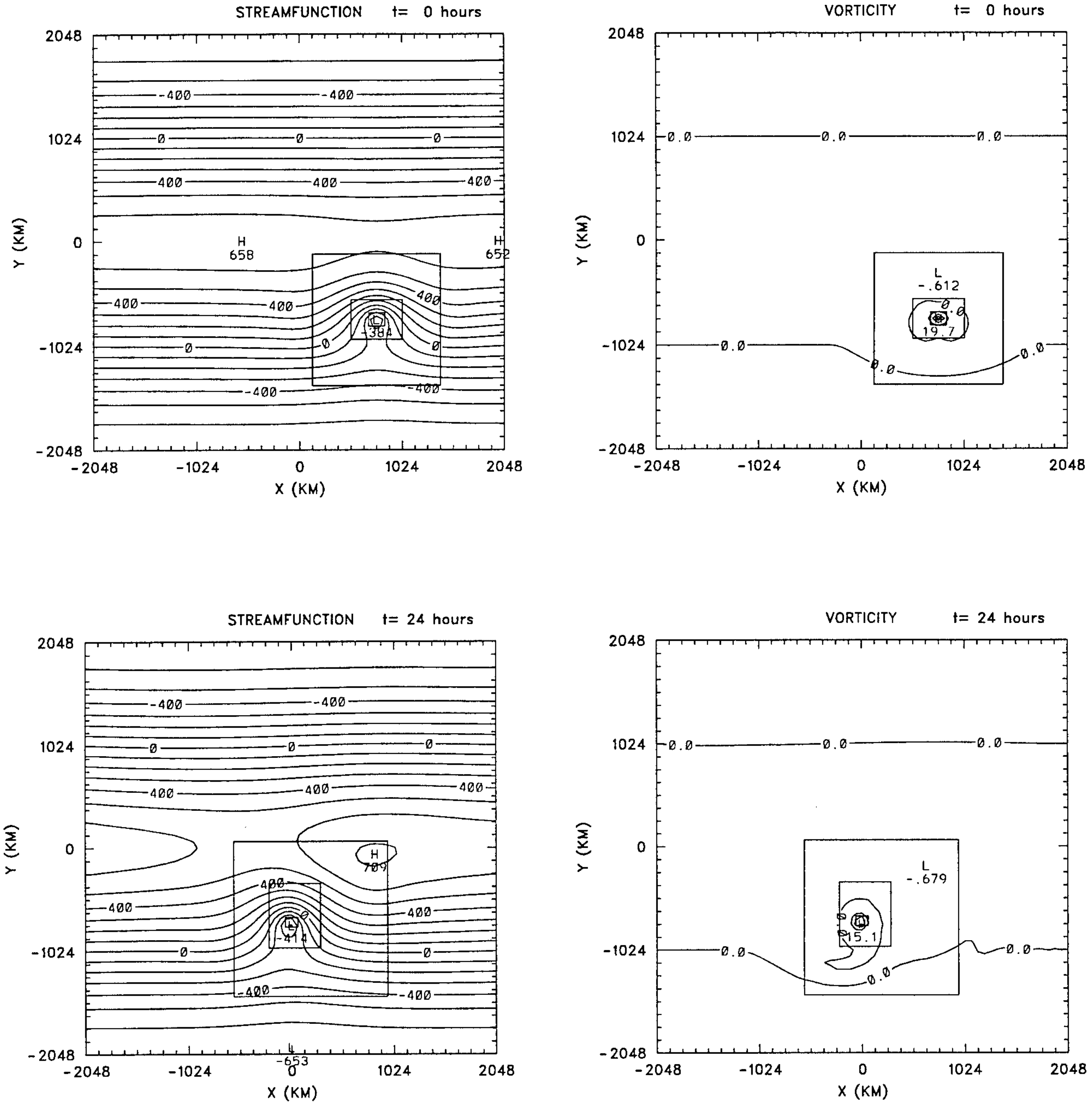

Figure 4: Sample solution: the weak hurricane case.

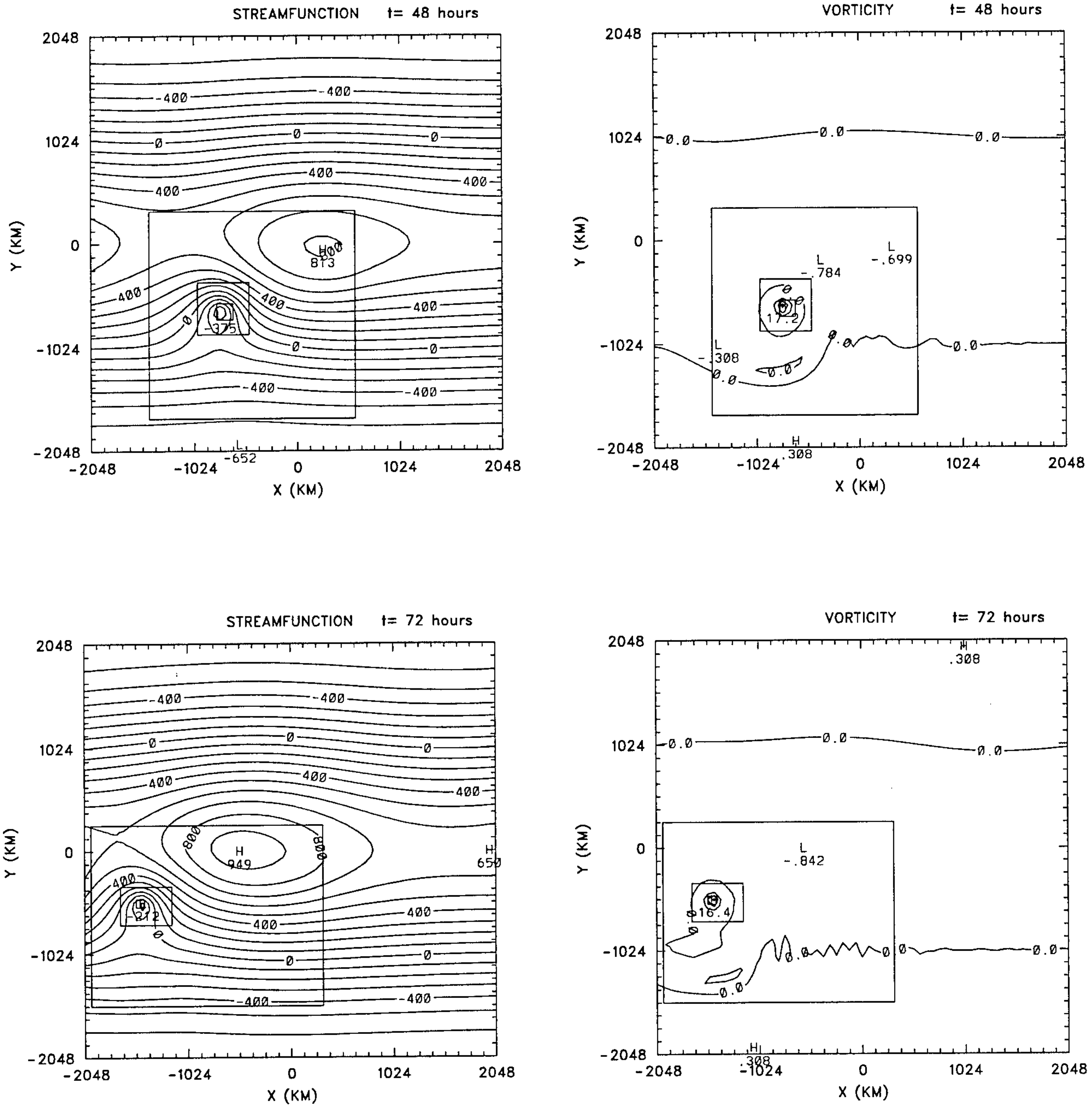

Figure 4 (continued).

4.3 Accuracy and efficiency

To measure the accuracy of the model in the absence of an analytical solution, we compare the tropical cyclone track to that obtained in a high-resolution reference run with uniform resolution $h = 4\,\text{km}$. The track error is defined as the distance between the location of the vortex center (vorticity maximum) and that of the reference run, with the center located by quadratic interpolation using four points surrounding the vorticity maximum on the finest grid (Smith et al., 1990); the mean forecast error is computed over a 72 hour model run (using trapezoidal quadrature with a step size of one hour). Computational work is measured by the CPU time required for the model run (on a SUN Ultra60 workstation with one 360 MHz processor).

Figure 5 shows the mean forecast error as a function of the computer time used for a large number of model runs for the weak hurricane case. The points plotted with "+" are for uniform-resolution model runs (i.e., a base grid with no finer grid patches) with the indicated resolution. The line joining these points indicates a baseline (uniform grid) performance. The isolated points to the left of this line are for model runs with various combinations of movable grids (i.e., fixed size patches), using essentially every reasonable combination of patches with mesh sizes from $h = 128\,\text{km}$ to $h = 4\,\text{km}$. The points plotted with "×" are for the adaptive model. These are connected by lines according to the finest mesh size allowed in the run; different points on each line correspond to different values of the exchange rate λ from 10^4 to 10^1, with errors generally decreasing—and CPU time increasing—as λ decreases. The figure shows a savings of roughly a factor of ten in computer time by using local mesh refinement, and suggests that in most cases the grids chosen by the adaptive algorithm are reasonable.

Figure 6 shows analogous results for the strong hurricane case. Since the vortex is both smaller and stronger, errors for a given amount of work are generally larger than in the previous case. Again the adaptive multigrid method chooses reasonable grids; in some cases the savings of execution time (compared to using uniform resolution) approach a factor of 100. Similar results are obtained in other cases. For example, for Figure 7 the strong hurricane vortex is embedded in a different environmental flow with zonal wind

$$\bar{u}(y) = \left(\frac{\bar{u}_0 y}{y_1}\right) \exp\left[\frac{1}{2}\left(1 - \frac{y^2}{y_1^2}\right)\right], \tag{17}$$

where $\bar{u}_0 = 10\,\text{m}\,\text{s}^{-1}$ and $y_1 = 443.4\,\text{km}$. For this case the vortex is centered initially at $x_0 = 768\,\text{km}$ and $y_0 = 768\,\text{km}$, where the positive Laplacian of the environmental vorticity leads to increased sensitivity of the vortex track to initial position errors (DeMaria, 1985). For the small vortex used here, this sensitivity is not evident; the adaptive model performs as well or better than in the other cases.

4.4 Effect of two-way interaction

Figure 8 demonstrates the effect of including two-way interaction between the computational grids. Each point represents the difference in mean forecast error between the Berger-Oliger

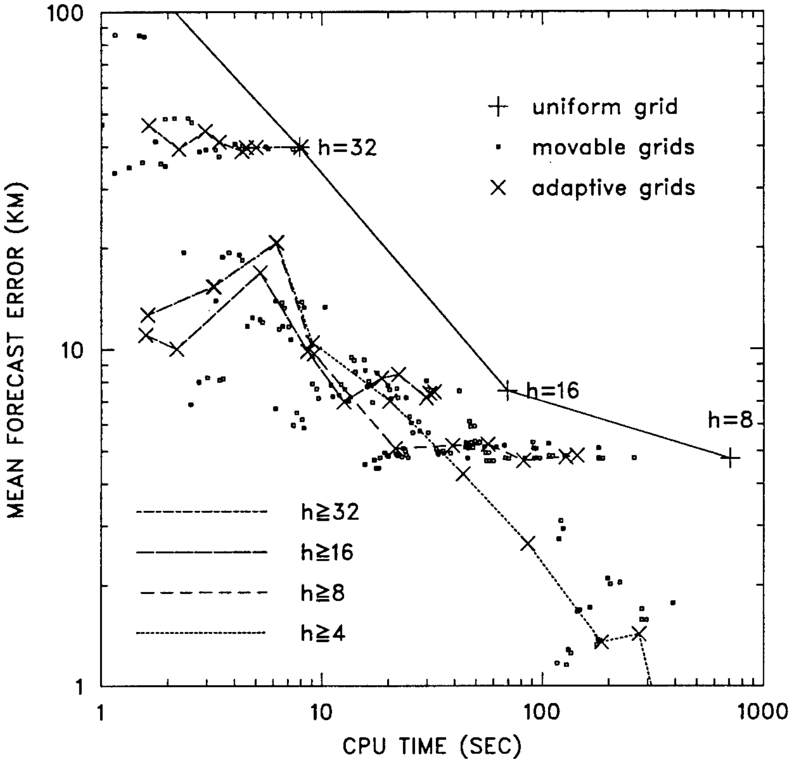

Figure 5: Error vs. CPU time for the weak hurricane case.

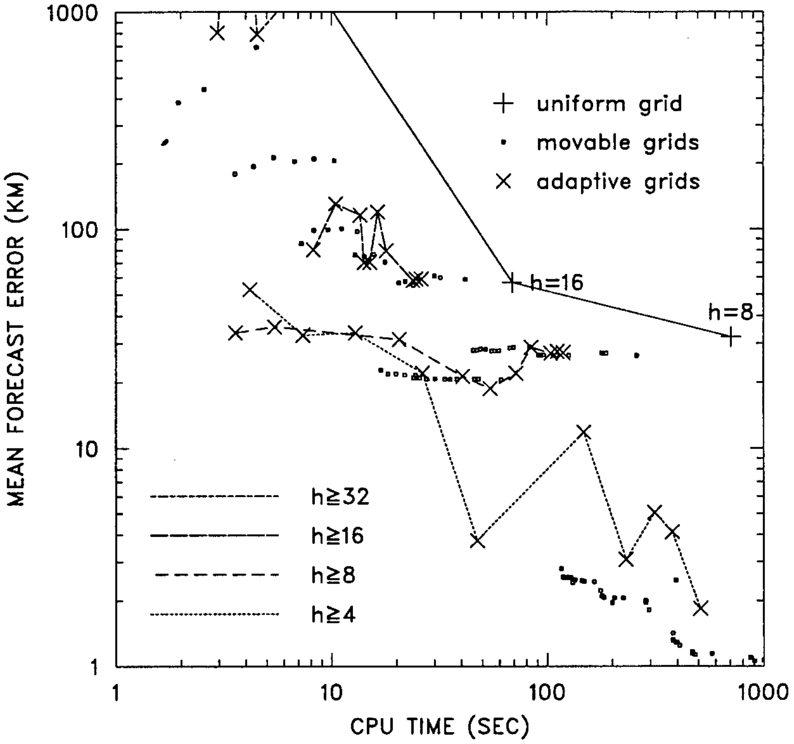

Figure 6: Error vs. CPU time the for strong hurricane case.

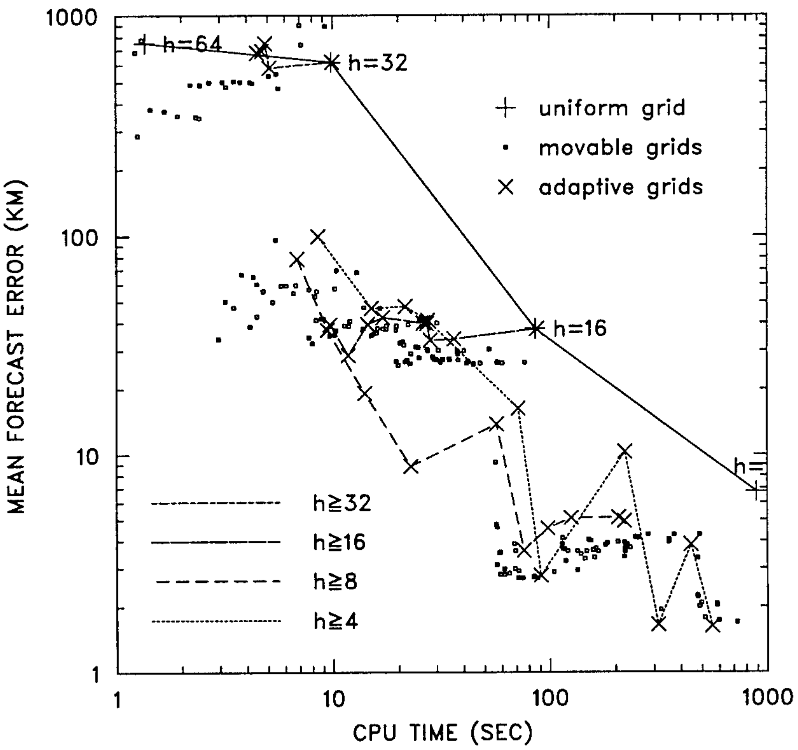

Figure 7: Same as Fig. 6 except for the environmental flow given by (17).

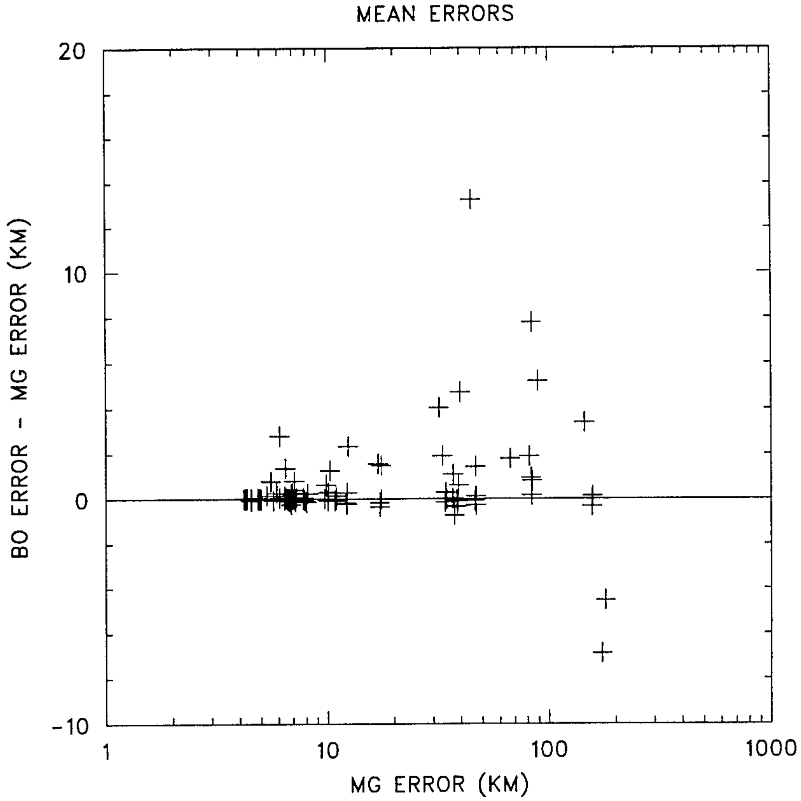

Figure 8: Comparison of accuracy achieved by the BO (one-way interaction) and MG (two-way interaction) methods for the weak hurricane case.

(BO) method and multigrid (MG) method as described in section 2.4. The points are plotted as a function of the error produced by the MG method for many runs with different choices of movable grids. The differences between the accuracy of the two methods are small—and can go either way—but when one method is better, it is usually the multigrid method, which includes two-way interaction.

4.5 A barotropically unstable vortex

The adaptive multigrid method might be especially advantageous for modeling a vortex which not only moves but has small-scale structure which evolves with time. While the nondivergent barotropic model employed here is incapable of representing the baroclinic effects and convection related to tropical cyclone intensification, it can represent the basic physical process of potential vorticity (PV) mixing. Thus, we consider the case of a barotropically unstable vortex which develops small-scale asymmetries due to chaotic nonlinear PV mixing and evolves toward a smooth symmetric vortex as it moves.

As an initial condition we use the "PV ring" vortex of Schubert et al. (1999) embedded in the zonal current (16). All parameters have the values given in that paper, except for the diffusion parameter ν for which we use the larger value $300\,\mathrm{m}^2\,\mathrm{s}^{-1}$, corresponding to an e-folding time of 338 s for a wavelength of 2 km. For this case we run the model with spherical geometry using the Mercator projection (1). We present results of two model runs: one using movable grids and the other using adaptive grid patches (with exchange rate $\lambda = 20$). In both cases we start with the base grid size 128×128 with mesh spacing $h = 32\,\mathrm{km}$ and corresponding time step $\Delta t = 20$ minutes. The movable grid run uses six square patches with side lengths ranging from 1/4 to 1/32 of the domain size. The adaptive run is limited to six adaptive grid patches; however, the model rejects the base grid as inadequate and consequently uses the base grid mesh spacing $h = 16\,\mathrm{km}$ and up to five adaptive grids. Thus, for both runs the model domain is 4096 km square and the finest mesh spacing is $h = 0.5\,\mathrm{km}$.

Figure 9 shows the details of the solution computed using movable grids (left panels) and adaptive grids (right panels). Only a region 256 km square centered on the vortex is shown. Since the vorticity is not always maximum at the center of the ring, we define the vortex center as the vorticity centroid (taken over the region where $\zeta > f$, to eliminate bias from the environmental vorticity). These results are quite similar to those of Schubert et al. (1999) up to at least $t = 6$ hours, verifying that the characteristic wavenumber-four pattern observed in that study is due to the dynamics, and is not an artifact of the periodicity required for the spectral model used. At later times, the solutions are qualitatively similar but differ in detail, as expected due to the chaotic nature of the PV mixing. Comparing the two solutions, we find that the details of the small-scale structure are quite similar; also, the vortex tracks (not shown) differ by only about 7 km over the course of the 72 hour model run. However, the run using adaptive grids was six times faster (about 16 minutes on a SUN Ultra60), since as the solution evolves toward a smooth symmetric vortex the finest grid patches are no longer needed and are automatically discarded.

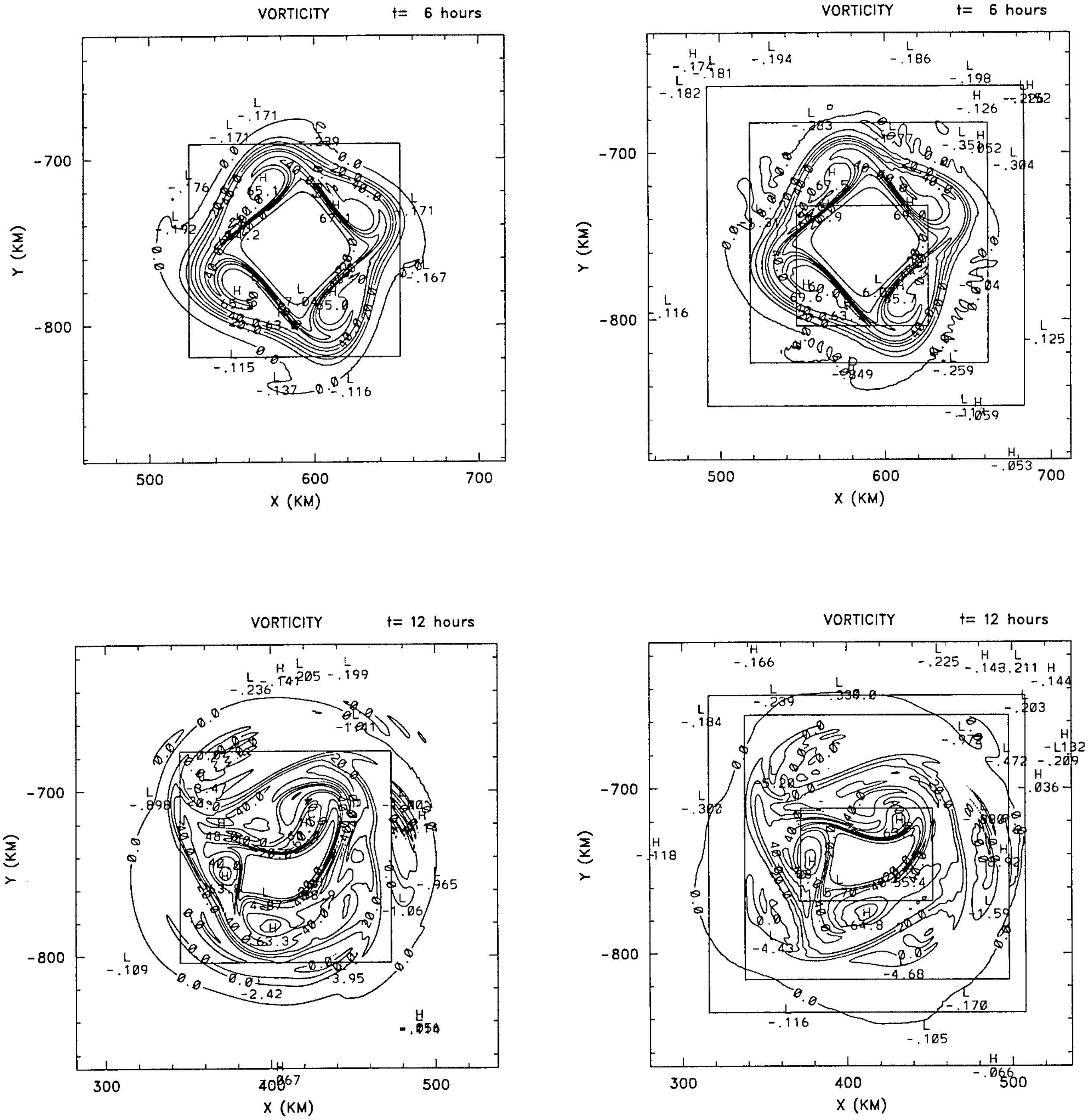

Figure 9: Vorticity for the PV ring case computed using movable grids (left panels) and adaptive grids (right panels). Solid rectangles show patch boundaries. The finest patch shown has mesh size $h = 0.5\,\text{km}$ in each case except the adaptive grids at $t = 24\,\text{hours}$ ($h = 1\,\text{km}$) and $t = 72\,\text{hours}$ ($h = 2\,\text{km}$).

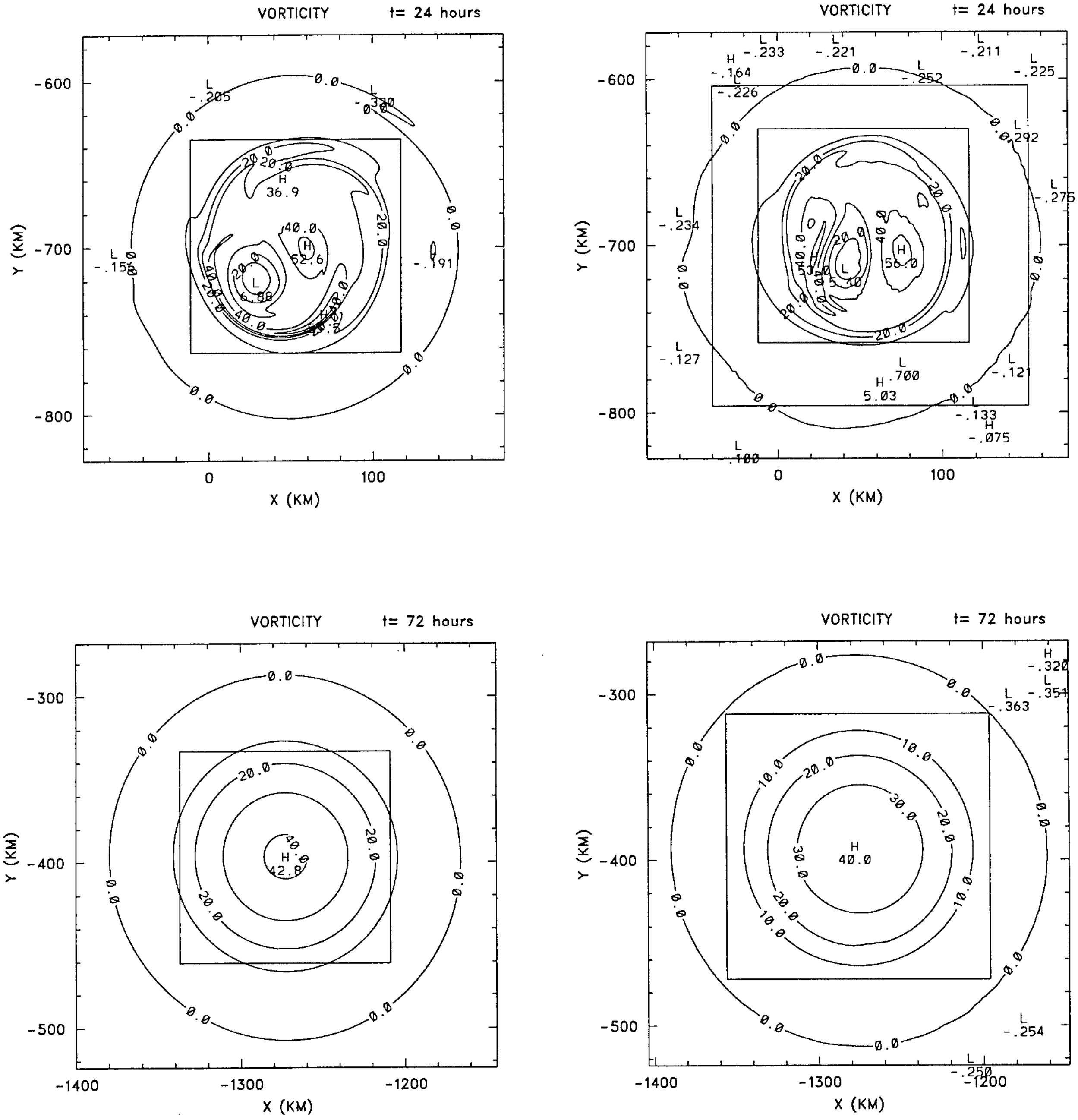

Figure 9 (continued).

5 Conclusions and Remarks

We have used the problem of tropical cyclone track prediction to investigate the potential of the adaptive multigrid method. Combining nesting of uniform grids, multigrid processing, and adaptive mesh refinement based on truncation error estimates, the method provides a seamless discretization with enhanced resolution precisely where it is needed. Numerical results from the nondivergent barotropic model MUDBAR show the method performs well, generally reducing the execution time by an order of magnitude or more (relative to using uniform grids) and choosing reasonable grids with a minimum of user intervention. Compared to conventional nested-grid models, the treatment of grid interfaces is particularly simple (although that simplicity may be more a result of using simple dynamics rather than the numerical method). These results are encouraging, and suggest that adaptive multigrid methods may be useful in other problems requiring time-dependent local mesh refinement.

While the model as it stands is fast and accurate, there is much room for improvement. We are currently working on two modifications which should substantially improve the inherent accuracy, namely, converting the model to the shallow-water equations and implementing fourth-order space differencing. To be useful for actual prediction, the model should be supplied with time-dependent boundary data from a global model and an appropriate initialization scheme. Given its speed and accuracy, the resulting model could be especially appropriate for ensemble forecasting techniques.

The numerical method presented here may in fact be overkill: for tropical cyclone track prediction, discretization errors may be less important than observational uncertainties or incomplete representations of the physics. Indeed, since the discretization is the only source of error considered here, our numerical results show far more accuracy than could be hoped for in operational prediction. Nevertheless, the adaptive multigrid approach allows one to effectively remove discretization error from the problem, and does so in a natural, robust, and efficient manner.

Acknowledgments

The helpful comments and advice of Mark DeMaria are gratefully acknowledged. Mark Loftis wrote the original (non-adaptive) version of the model, and Brittany Mitchell, Nicole Purick, and Hu Miao helped with the testing. This work was supported by the Office of Naval Research under grants N00014-98-1-0103 and N00014-98-1-0368.

References

Arakawa, A., 1966: Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. Part I. *J. Comput. Phys.*, **1**, 119–143.

Bai, D. and A. Brandt, 1987: Local mesh refinement multilevel techniques. *SIAM J. Sci. Stat. Comp.*, **8**, 109–134.

Berger, M. J. and J. Oliger, 1984: Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, **53**, 484–512.

Bernert, C., 1997: τ-extrapolation—theoretical foundation, numerical experiment, and application to Navier-Stokes equations. *SIAM J. Sci. Comp.*, **18**, 460–478.

Brandt, A., 1977: Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, **31**, 333–390.

Ciesielski, P. E., S. R. Fulton, and W. H. Schubert, 1986: Multigrid solution of an elliptic boundary value problem from tropical cyclone theory. *Mon. Wea. Rev.*, **114**, 797–807.

DeMaria, M., 1985: Tropical cyclone motion in a nondivergent barotropic model. *Mon. Wea. Rev.*, **113**, 1199–1210.

DeMaria, M., S. D. Aberson, K. V. Ooyama, and S. J. Lord, 1992: A nested spectral model for hurricane track forecasting. *Mon. Wea. Rev.*, **120**, 1628–1643.

Fulton, S. R., 1989: Multigrid solution of the semigeostrophic invertibility relation. *Mon. Wea. Rev.*, **117**, 2059–2066.

Fulton, S. R., 1997: A comparison of multilevel adaptive methods for hurricane track prediction. *Elec. Trans. Num. Anal.*, **6**, 120–132. `http://etna.mcs.kent.edu/`.

Fulton, S. R., P. E. Ciesielski, and W. H. Schubert, 1986: Multigrid methods for elliptic problems: A review. *Mon. Wea. Rev.*, **114**, 943–959.

Guinn, T. A. and W. H. Schubert, 1993: Hurricane spiral bands. *J. Atmos. Sci.*, **50**, 3380–3403.

Juang, H.-M. H. and J. E. Hoke, 1992: Application of fourth-order finite differencing to the NMC nested grid model. *Mon. Wea. Rev.*, **120**, 1767–1782.

Kurihara, Y. M. and M. A. Bender, 1980: Use of a movable nested-mesh model for tracking a small vortex. *Mon. Wea. Rev.*, **108**, 1792–1809.

Kurihara, Y. M., G. J. Tripoli, and M. A. Bender, 1979: Design of a movable nested-mesh primitive equation model. *Mon. Wea. Rev.*, **107**, 239–249.

Kurihara, Y. M., R. E. Tuleya, and M. A. Bender, 1998: The GFDL hurricane prediction system and its performance in the 1995 hurricane season. *Mon. Wea. Rev.*, **126**, 1306–1322.

Oliger, J. and A. Sundström, 1978: Theoretical and practical aspects of some initial boundary value problems in fluid dynamics. *SIAM J. Appl. Math.*, **35**, 419–446.

Ruge, J. W., S. F. McCormick, and S. Y. K. Yee, 1995: Multilevel adaptive methods for semi-implicit solution of shallow-water equations on a sphere. *Mon. Wea. Rev.*, **123**, 2197–2205.

Saleh, A. M., 1994: *Self-Adaptive Multilevel Methods for Fluid Flow Problems.* Ph.D. dissertation, Clarkson University.

Sanders, F., A. C. Pike, and J. P. Gaertner, 1975: A barotropic model for operational prediction of tracks of tropical storms. *J. Appl. Meteor.*, **14**, 265–280.

Schubert, W. H., M. T. Montgomery, R. K. Taft, T. A. Guinn, S. R. Fulton, J. P. Kossin, and J. P. Edwards, 1999: Polygonal eyewalls, asymmetric eye contraction, and potential vorticity mixing in hurricanes. *J. Atmos. Sci.*, **56**, 1197–1223.

Skamarock, W. C., 1989: Truncation error estimates for refinement criteria in nested and adaptive models. *Mon. Wea. Rev.*, **117**, 872–976.

Skamarock, W. C. and J. B. Klemp, 1993: Adaptive grid refinement for two-dimensional and three-dimensional nonhydrostatic atmospheric flow. *Mon. Wea. Rev.*, **121**, 788–804.

Smith, R. K., W. Ulrich, and G. Dietachmayer, 1990: A numerical study of tropical cyclone motion using a barotropic model. I: The role of vortex asymmetries. *Quart. J. Roy. Meteor. Soc.*, **116**, 337–362.

Appendix: Grid Size Adjustment

Once the desired grid size is determined based on the truncation error as described in section 3.2, it may be adjusted to ensure good coarsenability. For example, if the desired grid is 38×28 (grid intervals in x and y), this can be coarsened only once (to 19×14) for multigrid processing; since the coarsest grid is large, the multigrid solver is inefficient. In such a situation it will pay to increase the grid size slightly, e.g., to 40×28 (which can be coarsened twice down to 10×7) or even to 40×32 (which can be coarsened three times down to 5×4). We do this as follows. Starting with the desired size $N_x \times N_y$ (grid intervals in x and y):

1. Search for a grid slightly *larger* than desired by increasing N_x and/or N_y so they have at least one more factor of two in common, thus allowing at least one more coarse grid for multigrid processing. Accept the larger grid only if it can be strictly nested and gives a predicted gain in efficiency. Repeat until no gain in efficiency is produced.

2. Test grid sizes slightly *smaller* than desired (while keeping at least as many coarse grid levels as obtained in step 1). Accept such a grid only if it results in a substantial predicted gain in efficiency.

To predict the gain in efficiency for a proposed change of grid size, let $W_\zeta(N_x, N_y)$ and $W_\psi(N_x, N_y)$ represent the computational work required to predict ζ and solve for ψ, respectively, for one time step on a computational grid of size $N_x \times N_y$. We assume that $W_\zeta(N_x, N_y)$ is proportional to the number of grid points, and thus to the product $N_x N_y$. We also assume—at least for grids which are sufficiently coarsenable—that $W_\psi(N_x, N_y) \approx \omega W_\zeta(N_x, N_y)$ for some constant factor ω; empirically we find that $\omega \approx 2$. Then the gain in efficiency (i.e., the speedup) in changing from grid size $N_x \times N_y$ to $\hat{N}_x \times \hat{N}_y$ is predicted to be

$$\frac{W_\zeta(N_x, N_y) + W_\psi(N_x, N_y)}{W_\zeta(\hat{N}_x, \hat{N}_y) + W_\psi(\hat{N}_x, \hat{N}_y)} \approx \frac{1}{\omega + 1}\left[\omega \frac{W_\psi(N_x, N_y)}{W_\psi(\hat{N}_x, \hat{N}_y)} + \frac{N_x N_y}{\hat{N}_x \hat{N}_y}\right]. \tag{18}$$

To estimate W_ψ we first estimate the work required for a multigrid V-cycle with m grid levels (using ν_c relaxation sweeps on the coarsest grid and ν_f on each of the finer grids) as

$$W_V(m) = \nu_f \left(1 + \frac{1}{4} + \frac{1}{16} + \cdots + \frac{1}{4^{m-2}}\right) + \nu_c = \frac{\nu_f}{3}\left(4 - \frac{1}{4^{m-2}}\right) + \frac{\nu_c}{4^{m-1}} \tag{19}$$

work units (one work unit is the work of one sweep on the finest grid), where we have neglected the work of grid transfers. The corresponding work for the FMG algorithm with one V-cycle per level is then

$$W_F(m) = W_V(m) + \frac{1}{4} W_V(m-1) + \cdots + \frac{1}{4^{m-2}} W_V(2) + \frac{\nu_c}{4^{m-1}} \tag{20}$$

work units. The number of sweeps on the coarsest grid is set by the requirement that $(\rho_G)^{\nu_c} \approx (\bar{\mu})^{\nu_f}$, where

$$\rho_G = \frac{1}{4}\left[\cos\left(\frac{2^{m-1}\pi}{N_x}\right) + \cos\left(\frac{2^{m-1}\pi}{N_y}\right)\right]^2 \tag{21}$$

is the spectral radius of the Gauss-Seidel iteration operator on the coarsest grid and $\bar{\mu} = 0.25$ is the multigrid smoothing factor for Gauss-Seidel relaxation with red-black ordering. Solving for ν_c, substituting from (19) in (20) and multiplying by the number of grid points on the finest grid yields the estimate

$$W_\psi(N_x, N_y) = \frac{N_x N_y}{4^{m-1}} \left[\frac{16}{9}(4^{m-1} - 1) - \frac{4}{3}(m-1) + m\frac{\log(\bar{\mu})}{\log(\rho_G)} \right\} \qquad (22)$$

for the total number of operations required to solve for ψ on a grid of size $N_x \times N_y$ (here normalized by the number of operations required per gridpoint per sweep). Combining (22) with (18) gives an effective way to predict the gain in efficiency when considering a new grid size.